

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-013399

(43)Date of publication of application : 14.01.2000

(51)Int.Cl.

H04L 12/28

G06F 13/00

G06F 13/14

G06F 13/38

(21)Application number : 10-179330

(71)Applicant : RICOH CO LTD

(22)Date of filing : 25.06.1998

(72)Inventor : WATANABE HIDEYUKI

(54) IDENTIFICATION METHOD FOR NETWORK DEVICE

(57)Abstract:

PROBLEM TO BE SOLVED: To avoid occurrence of a case where the network devices are connected to another network of the same type by allocating the device identification tags to the network devices and identifying these devices of the same type based on the identification tags.

SOLUTION: When the production of a tag is started, the type of a network device is first retrieved in a local network. The value of a device identification tag TAG is stored even when the power supply of the network device is cut and never changed as long as it's set again. When connecting two scanners, if one of the two scanners is already connected to an IEEE1394 network and the other scanner is to be added the value of the TAG is set at 1 if the latter scanner is once used. As each scanner has different TAGs, the TAGs of both scanners never change.

LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2000 Japan Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2000-13399
(P2000-13399A)

(43) 公開日 平成12年1月14日 (2000.1.14)

(51) Int.Cl. ⁷	識別記号	F I	テーマコード [*] (参考)
H 0 4 L 12/28		H 0 4 L 11/00	3 1 0 D 5 B 0 1 4
G 0 6 F 13/00	3 5 7	G 0 6 F 13/00	3 5 7 A 5 B 0 7 7
	13/14		13/14 3 2 0 A 5 B 0 8 9
	13/38		13/38 3 5 0 5 K 0 3 3

審査請求 未請求 請求項の数 6 O L (全 11 頁)

(21) 出願番号 特願平10-179330

(22) 出願日 平成10年6月25日 (1998.6.25)

(71) 出願人 000006747

株式会社リコー

東京都大田区中馬込1丁目3番6号

(72) 発明者 渡辺 英行

東京都大田区中馬込1丁目3番6号 株式会社リコー内

(74) 代理人 100060690

弁理士 瀧野 秀雄

Fターム(参考) 5B014 HB01

5B077 AA03

5B089 DD03

5K033 AA05 BA15 DB12 DB14 EA07

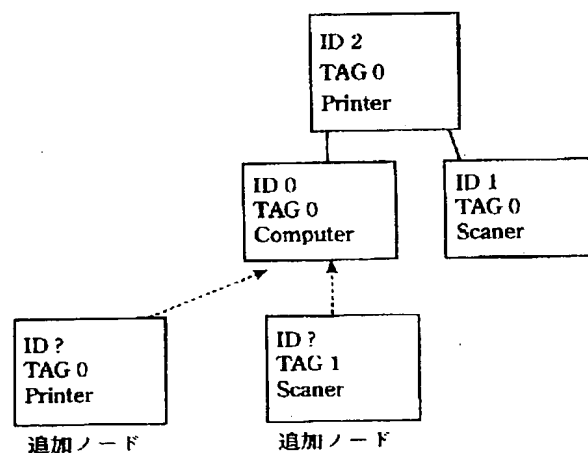
EC03

(54) 【発明の名称】 ネットワーク機器識別方法

(57) 【要約】

【課題】 I D再設定前後でI Dが変更されるようなケースを回避した一元的なI D管理を実現すること。

【解決手段】 ネットワーク上に接続されている複数台の同種類のネットワーク機器を識別する識別方法において、所定のカテゴリ内での同種類のネットワーク機器を識別するための機器識別用タグをネットワーク機器の各々に割り当てると共に、機器識別用タグに基づいて同種類のネットワーク機器の個々を識別する。



【特許請求の範囲】

【請求項 1】 ネットワーク上に接続されている複数台の同種類のネットワーク機器を識別する識別方法において、

所定のカテゴリ内での同種類のネットワーク機器を識別するための機器識別用タグを当該ネットワーク機器の各々に割り当てると共に、当該機器識別用タグに基づいて当該同種類のネットワーク機器の個々を識別することを特徴とするネットワーク機器識別方法。

【請求項 2】 前記機器識別用タグは、所定のトリガ信号がネットワーク内で発生したとき当該トリガ信号に応じてネットワーク機器の各々がネットワーク上で相互に交信して前記カテゴリ内でユニークに設定されることを特徴とする請求項 1 に記載のネットワーク機器識別方法。

【請求項 3】 前記機器識別用タグの再設定は、前記トリガ信号が新たに発生しない限り実行されないことを特徴とする請求項 1 に記載のネットワーク機器識別方法。

【請求項 4】 2つのネットワークを相互接続する場合であって、当該2つのネットワークにおいて同種類のネットワーク機器間で前記機器識別用タグが同一であるようなコンフリクトが発生したとき、当該コンフリクトしているネットワーク機器に対する前記機器識別用タグの再設定を実行することを特徴とする請求項 1 に記載のネットワーク機器識別方法。

【請求項 5】 ネットワークに追加接続されたネットワーク機器に対して前記機器識別用タグの再設定を行うことを特徴とする請求項 4 に記載のネットワーク機器識別方法。

【請求項 6】 前記ネットワークに追加接続されたネットワーク機器とは、機器の電源リセット以降に前記機器識別用タグを再設定していないネットワーク機器であることを特徴とする請求項 5 に記載のネットワーク機器識別方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、ネットワーク機器間のデータ転送に関するものであり、特に、ネットワーク上に同種類のネットワーク機器が少なくとも2個以上あるときに、同種類のネットワーク機器の識別に関する。

【0002】 具体的には、ネットワーク上の機器の識別、IEEE 1212 に準拠したアドレス構成を持つネットワークに適用でき、特に、IEEE 1394 に準拠したネットワーク機器の識別に適用できる。

【0003】

【従来の技術】 従来この種のネットワーク機器識別方法としては、例えば、IEEE (米国電気電子技術者協会) が規格化した新しい高速シリアルインタフェースである IEEE 1394 がある。

【0004】 IEEE 1394 に準拠したネットワーク機器は、バスリセットを検知した時には自動的に識別子 (ID) が付加されるが、その後に再度のバスリセットが起こったときは、このネットワーク機器に再度のバスリセット以前の ID と同じ ID が付加される保証は定義されていないため、再度のバスリセットの前後で ID が変化してしまう可能性があった。従って、IEEE 1394 に準拠したネットワーク上に同種類 (例えば、プリンタ、スキャナー、ハードディスク、パソコン (PC)、ビデオカメラなど) のネットワーク機器が少なくとも2個以上ある場合に新たにバスリセットが発生すると ID が変更される可能性があるため、このバスリセット以前に使用してきたネットワーク機器が使用できなくなる可能性があり、また別の同種類のネットワークに接続される可能性がある。

【0005】 前述のような ID 管理のみでは、バスリセット後に ID が変化する可能性があるため、ネットワーク機器の統一された ID を用いた管理が難しい。このため IEEE 1394-1995 の規格書では、コンフィグレーション (configuration) ROM で 64 ビット構成の `node_vendor`, `chip_id_hi`, `chip_id_lo` といった値が重複 (コンフリクト) しないように、ネットワークを供給するベンダー側で ID の一元的な管理しなければならないと規定している。

【0006】

【発明が解決しようとする課題】 しかしながら、ネットワーク上に同種類のネットワーク機器を複数台接続した場合、ネットワークを供給するベンダー毎に ID 管理の方法が異なるケースが一般的であると考えられ、ベンダーが異なるネットワーク間を相互接続するような場合、一方のネットワークに接続されていたネットワーク機器を外して他方のネットワークに接続するような場合、あるいはバスリセットが随時発生するような場合には ID の再設定が実行されるため、やはり ID 再設定前後で ID が変更される可能性があり、この ID 再設定以前に使用してきたネットワーク機器が使用できなくなる可能性があり、また別の同種類のネットワークに接続される可能性があるといった問題が依然として解決されていない。

【0007】 本発明は、このような従来の問題点を解決することを課題としており、特に、ネットワーク上に同種類のネットワーク機器を複数台接続する際、ベンダーが異なるネットワーク間を相互接続するような場合、一方のネットワークに接続されていたネットワーク機器を外して他方のネットワークに接続するような場合、あるいはバスリセットが随時発生するような場合に、ID 再設定前後で ID が変更されるようなケースを回避した一元的な ID 管理を実現し、ID 再設定以前に使用してきたネットワーク機器が使用できなくなるケース、別の同

種類のネットワークに接続されるケースを回避できるネットワーク機器の識別方法を提供することを目的としている。

【0008】

【課題を解決するための手段】上記課題を解決するため成された請求項1に記載の発明は、ネットワーク上に接続されている複数台の同種類のネットワーク機器を識別する識別方法において、所定のカテゴリ内での同種類のネットワーク機器を識別するための機器識別用タグを当該ネットワーク機器の各々に割り当てると共に、当該機器識別用タグに基づいて当該同種類のネットワーク機器の個々を識別するネットワーク機器識別方法である。

【0009】請求項1に記載の発明によれば、所定のカテゴリ内での同種類のネットワーク機器を識別するための機器識別用タグに基づいて同種類のネットワーク機器の個々を識別するので、ネットワーク上に同種類のネットワーク機器を複数台接続する際、ベンダーが異なるネットワーク間を相互接続するような場合、一方のネットワークに接続されていたネットワーク機器を外して他方のネットワークに接続するような場合、あるいはバスリセットが随時発生するような場合であっても、ID再設定前後でIDが変更されるようなケースを回避した一元的なID管理を実現できるようになり、その結果、ID再設定以前に使用してきたネットワーク機器が使用できなくなるケース、別の同種類のネットワークに接続されるケースを回避できるようになる。

【0010】また請求項2に記載の発明は、請求項1に記載のネットワーク機器識別方法において、前記機器識別用タグは、所定のトリガ信号がネットワーク内で発生したとき当該トリガ信号に応じてネットワーク機器の各々がネットワーク上で相互に交信して前記カテゴリ内でユニークに設定されるネットワーク機器識別方法である。

【0011】請求項2に記載の発明によれば、請求項1に記載の効果に加えて、カテゴリ内でユニークに設定される機器識別用タグに基づいて、一元的なID管理を実現できるようになる。

【0012】また請求項3に記載の発明は、請求項1に記載のネットワーク機器識別方法において、前記機器識別用タグの再設定は、前記トリガ信号が新たに発生しない限り実行されないネットワーク機器識別方法である。

【0013】請求項3に記載の発明によれば、請求項1に記載の効果に加えて、トリガ信号の発生に応じたタイミングで一元的なID管理を実現できるようになる。

【0014】また請求項4に記載の発明は、請求項1に記載のネットワーク機器識別方法において、2つのネットワークを相互接続する場合であって、当該2つのネットワークにおいて同種類のネットワーク機器間で前記機器識別用タグが同一であるようなコンフリクトが発生したとき、当該コンフリクトしているネットワーク機器に

対する前記機器識別用タグの再設定を実行するネットワーク機器識別方法である。

【0015】請求項4に記載の発明によれば、請求項1に記載の効果に加えて、コンフリクトを解除するような一元的なID管理を実現できるようになる。

【0016】また請求項5に記載の発明は、請求項4に記載のネットワーク機器識別方法において、ネットワークに追加接続されたネットワーク機器に対して前記機器識別用タグの再設定を行うネットワーク機器識別方法である。

【0017】請求項5に記載の発明によれば、請求項4に記載の効果に加えて、ネットワークに追加接続されたネットワーク機器に対する一元的なかつ限定的なID管理を実現できるようになる。

【0018】また請求項6に記載の発明は、請求項5に記載のネットワーク機器識別方法において、前記ネットワークに追加接続されたネットワーク機器とは、機器の電源リセット以降に前記機器識別用タグを再設定していないネットワーク機器とするネットワーク機器識別方法である。

【0019】請求項6に記載の発明によれば、請求項5に記載の効果に加えて、電源リセットの発生に応じたタイミングで一元的なID管理を実現できるようになる。

【0020】

【発明の実施の形態】以下の各実施形態では、ネットワークの実施形態として、IEEE（米国電気電子技術者協会）が規格化した新しい高速シリアルインタフェースであるIEEE1394-1995の通信プロトコルに準拠したネットワーク（IEEE1394ネットワーク）を用いることにするが、これに特に限定されことなく、シリアルデータ通信ができ、ノードであるネットワーク機器がツリー状に接続されているネットワークであれば、広く適用可能である。

【0021】以下の各実施形態は、IEEE1394ネットワーク上に接続されている複数台の同種類のネットワーク機器を識別する識別方法であって、所定のカテゴリ内での同種類のネットワーク機器を識別するための機器識別用タグTAGをネットワーク機器の各々に割り当てると共に、機器識別用タグTAGに基づいて同種類のネットワーク機器の個々を識別する点に特徴を有している。ここでネットワーク機器としては、AV機器、プリンタ、デジタルカメラ、パソコン等が考えられる。

【0022】また、第1実施形態及び第2実施形態のネットワーク機器識別方法は、IEEE1394-1995のプロトコルスタックで定義されている物理層（PHYレイヤ）、リンクレイヤ、BM（Bus Manager）、トランザクションレイヤなどにインプリメントされて実行されている。ここで、PHYレイヤやリンクレイヤは、シリコンでハードウェアで構成されており、BMやトランザクションレイヤは、ソフトウェアで構成

されている。

【0023】このため、各ネットワーク機器は、PHYレイヤを介してIEEE1394ネットワークに接続された状態で、パケットの送受信（例えば、アイソクロナス転送やアシンクロナス転送）を他のネットワーク機器との間で送受信することになる。

【0024】（第1実施形態）図1、図2を用いて第1実施形態について説明する。今、図1に示すように、機器識別用タグTAGの異なるスキャナ（Scanner（ノードID？、TAG#1）、ネットワーク機器の一形態）と、機器識別用タグTAGが同じプリンタ（Printer（ノードID？、TAG#0）、ネットワーク機器の一形態）を、追加ノードとしてIEEE1394で規定されているネットワーク（IEEE1394ネットワーク）に接続しようとし、その結果、図2に示すように、Scanner（ノードID#1、TAG#1）とPrinter（ノードID#0、TAG#0）をIEEE1394ネットワークに接続したので、各機器のノードIDが変化している。具体的には、Scanner（ノードID#1、TAG#0）→Scanner（ノードID#3、TAG#0）、コンピュータ（Computer（ノードID#0、TAG#0）、ネットワーク機器の一形態）→Computer（ノードID#2、TAG#0）、Printer（ノードID#2、TAG#0）→Printer（ノードID#4、TAG#1）となるノードID再設定を行うノードID管理について考える。

【0025】本実施形態では、機器識別用タグTAGの再設定は、バスリセットが新たに発生しない限り実行されないようにしている。また機器識別用タグTAGは、再度設定されない限り変更されないように、あるいはネットワーク機器の電源が遮断されても保存されるように設定している。これにより、バスリセットの発生に応じたタイミングで一元的なノードID管理を実現できるようになる。

【0026】また本実施形態では、2つのIEEE1394ネットワークを相互接続する場合であって、2つのIEEE1394ネットワークにおいて同種類のネットワーク機器間で機器識別用タグTAGが同一であるようなコンフリクトが発生したとき、コンフリクトしているネットワーク機器に対する機器識別用タグTAGの再設定を実行するように設定している。これにより、コンフリクトを解除するような一元的なノードID管理を実現できるようになる。

【0027】また本実施形態では、IEEE1394ネットワークに追加接続されたネットワーク機器に対して機器識別用タグTAGの再設定を行うように設定している。これにより、IEEE1394ネットワークに追加接続されたネットワーク機器に対する一元的なかつ限定的なノードID管理を実現できるようになる。

【0028】ここで、IEEE1394ネットワークに追加接続されたネットワーク機器とは、機器の電源リセット以降に機器識別用タグTAGを再設定していないネットワーク機器としている。これにより、電源リセットの発生に応じたタイミングで一元的なノードID管理を実現できるようになる。

【0029】具体的に説明すると、機器識別用タグTAGの値は、IEEE1212で規定されているコンフィグレーション（configuration）ROM中のチップノードIDに格納されている。本実施形態では、特に、IEEE1212のCSRアーキテクチャに規定されている制御用レジスタであるControl Status Register（CSR）のコンフィグレーション（configuration）ROMに格納されている。

【0030】このようなコンフィグレーションROM内には、Node_Unique_ID、Bus_Info_Block中のNode_Vendor_ID（24ビット）、chip_id_hi（8ビット）、chip_id_lo（32ビット）等のフィールドが設けられている。

【0031】図3は、第1実施形態のネットワーク機器識別方法の動作を説明するためのフローチャートを示す。

【0032】ステップS100でタブの作成を開始すると、まず、ステップS110の処理でローカルネットワーク上のネットワーク機器の種類（Printer、Scanner、Computer）を検索する。

【0033】ネットワーク機器の検索では、CSR中のコンフィグレーションROM内のデータフィールドであるNode_Unique_IDまたは、Bus_Info_Block中のNode_Vendor_ID（24ビット）、chip_id_hi（8ビット）、chip_id_lo（32ビット）が同じものを検索する（図4参照）。

【0034】コンフィグレーションROMは、各ネットワーク機器のメモリ空間内のレジスタ空間内に保持されている。レジスタ空間には、その他に、CSRアーキテクチャに準拠したCSRコアレジスタ、シリアルバスレジスタ、ユニット空間レジスタ用のメモリ空間が用意されている。

【0035】第1実施形態で用いる機器識別用タグTAGは、図5、図6、図7に示すように、Node_Unique_IDリーフ内のchip_id_hi（8ビット）またはchip_id_lo（32ビット）に実装可能である。

【0036】ここで、機器識別用タグTAGの値は、ネットワーク機器の電源が遮断されていても保存されており、設定されない限り変更されないようになっている。

50 具体的には、バックアップ電源で保存されているかまた

は、不揮発性RAM上に保存されている。

【0037】図5、図7のchip_id_hiに機器識別用タグTAGを実装した場合には、Node_Vendor_IDとchip_id_loを検索キーとしてネットワーク機器を検索する。

【0038】一方、図6のchip_id_loに機器識別用タグTAGを実装した場合には、Node_Vendor_IDとchip_id_hiの内容を検索キーとしてネットワーク機器を検索する。

【0039】次に、ステップS200の処理では、同じ種類のネットワーク機器の機器識別用タグTAGを自動的に設定する。ステップS200をすべてのネットワーク機器の種類に対して実行する。

【0040】図8のフローチャートを用いて、ステップS200を更に詳しく述べる。

【0041】ステップS200に入る時点で、IEEE1394ネットワークの種類(図8では"タイプ"と記述してある)がx(エックス)と指定されている。最初にタイプxのネットワーク機器の個数Nを検索する(ステップS210)。

【0042】ステップS210で検索されたネットワーク機器の個数Nが1のとき(ステップS220のYES)は、(図2の例ではComputer装置)、同じ種類のネットワーク機器が存在しない。従って、検索されたネットワーク機器の機器識別用タグTAG(x, 0)の値を0とする(ステップS221)。

【0043】図1に示すように2台のScannerを接続するケースにおいて、一方のScannerがIEEE1394ネットワーク上に既に接続されており、他方のScannerを追加接続しようとしている場合、追加接続しようとしているScanner(ノードID?, TAG#1)はすでにIEEE1394ネットワーク上で使用されたことがあり、機器識別用タグTAGの値が1に設定されている。各Scannerの機器識別用タグTAGはすべて異なるので(ステップS230のYES)、これら2台のScannerの機器識別用タグTAGは変化しない。すなわち、Scanner(ノードID1, TAG#0)→Scanner(ノードID3, TAG#0)であり、Scanner(ノードID?, TAG#1)→Scanner(ノードID1, TAG#1)となる。

【0044】一方、図1に示すPrinter(ノードID?, TAG#0)は、IEEE1394ネットワーク上に初めて接続しようとしている。従って、2台のPrinterは機器識別用タグTAGの値が同じ(両方ともTAG#0)であるので、機器識別用タグTAG(x, i)の付け直し処理(ステップS240)を行う。具体的には、最初に見つかったPrinterの機器識別用タグTAGの値を0とする。次に見つかったPrinterの値は先に見つかった機器識別用タグTAG

Gの値より1だけ大きい値に設定する。

【0045】具体的には、Printer(ノードID2, TAG#0)→Printer(ノードID4, TAG#1)、Printer(ノードID?, TAG#0)→Printer(ノードID0, TAG#0)となる。こうして、Printerの接続数だけ機器識別用タグTAGの値が増える。

【0046】図9に本発明における機器識別用タグTAGの設定方法の他の実施形態のネットワーク構成図を示す。図10、図11は、バスリセットによる機器識別用タグの設定の動作を説明するためのフローチャートである。

【0047】図10、図11のフローの上位フローには、図8に示す前述のネットワーク識別フローの例と同様に、図3のネットワーク識別フローを使用することができる。

【0048】図10に示すIEEE1394ネットワークの識別フローで使用する機器識別用タグTAGは、図7で示したchip_id_hiへの実装に限るものではなく、chip_id_loにも実装可能である。

【0049】本実施形態の機器識別用タグTAGの値は、ネットワーク機器の電源が遮断されてもバックアップ電源もしくは不揮発性RAMでバックアップされている。図7に示すEntryFlag(1ビット)とCheckFlag(1ビット)は、ネットワーク機器の電源がオンになったとき0(ゼロ)に初期化(リセット)される。

【0050】また機器識別用タグTAGは、所定のトリガ信号がIEEE1394ネットワーク内で発生したときトリガ信号に応じてネットワーク機器の各々がIEEE1394ネットワーク上で相互に通信してカテゴリ内でユニークに設定されるように設定している。

【0051】具体的には、同種類のネットワーク機器の機器識別用タグTAGは、IEEE1394ネットワーク内でバスリセット(トリガ信号)が発生後に設定するようにしている。バスリセットは、IEEE1394に規定されているように、ネットワーク機器の初期電源投入時、動作中に他のノードが接続された時、あるノードが切り離された時、他のIEEE1394ネットワークが接続されて一体化したバスが形成された時、2つのIEEE1394ネットワークに分割された時、ブランチが切り離された時、接続中のノードの電源がONまたはOFFされた時等、つまりネットワークのツリー構造に何らかの変化が起こったことが検出されたときに生じられる特性を持っている。なお、IEEE1394-1995の規格書には、バスリセットに応じてコンフィグレーション機能が生じられることも規定されている。

【0052】これにより、カテゴリ内でユニークに設定される機器識別用タグTAGに基づいて、一元的なノードID管理を実現できるようになる。

【0053】図9のネットワーク構成例及び前述の図10、図11のフローチャートに基づいて、このようなバスリセットによる機器識別用タグTAGの設定の動作を説明する。

【0054】図9は、3台のIEEE1394インタフェース機能を有するネットワーク機器（例えば、図中の1台のComputerと2台のPrinter）の電源が入り、ネットワーク機器がIEEE1394ネットワークに接続されて、バスリセットが終了した状態を示している。

【0055】ノードIDはIEEE1394ネットワークのノードIDである。TAG、EntryFlag、CheckFlagの値は、前述の図7の値を示しており、ネットワーク機器の電源が入って、バスリセットが終了した状態なので、EntryFlagとCheckFlagの値は0（ゼロ）になっている。

【0056】まず、IEEE1394ネットワークの種類としてComputerが選択されたとする。このとき、EntryFlagとCheckFlagの値は0（ゼロ）になっているので、ステップS330の処理まで進む。

【0057】ネットワーク機器の種類がComputerである場合で、かつ機器識別用タグTAGの値が0のノードはIEEE1394ネットワーク上に存在しない（ステップS340のNO）ので、ステップS350の処理まで進み、EntryFlagの値が1に設定される。

【0058】さらにステップS350に続くステップS360の処理では、機器識別用タグTAGの記述されているタグマップに機器識別用タグTAG#0が登録される。

【0059】ステップS360に続くステップS370の処理では、CheckFlagの値を1に設定してノードが「検査済み」の状態になる。

【0060】さらにステップS370に続くステップS310の処理では、EntryFlagの値が0のノードはない（ステップS310のNO）ので、処理が図11に示すステップS390の処理に進む。

【0061】このときタグキューは空（ステップS390のYES）なので、ComputerというIEEE1394ネットワークの種類の機器識別用タグTAGの設定処理は終了する。

【0062】続いて、機器識別用タグTAGの記述されているタグマップについて説明する。

【0063】機器識別用タグTAGの記述されているタグマップは、図12に示す機器識別用タグTAGの記述されている64ビット長のタグマップレジスタに記述されている。ここで、TagMapReg_loのLSB（Least Significant Bit）で機器識別用タグTAG#0を、MSB（Most Sig

nificant Bit）で機器識別用タグTAG#31が登録済みかどうかを判定できる。またTagMapReg_hiのLSBで機器識別用タグTAG#32を、MSB-1（MSBの隣のビット）で機器識別用タグTAG#62が登録済みかどうかを判定できる。バスリセット後は、この機器識別用タグTAGの記述されているマップレジスタの値は0に設定されており、すべての機器識別用タグTAGが未使用となっている。このマップレジスタはネットワーク機器の種類ごとに用意されている。

【0064】次にPrinterというネットワーク機器の機器識別用タグTAGを設定する動作を、図10及び図11に基づいて説明する。

【0065】ステップS310において、EntryFlagの値が0であるノードを検索する。図9のIEEE1394ネットワークではPrinterが2個あるが、ノードIDが小さい方（すなわち、ノードIDが1のノード）から検索が開始される。

【0066】ノードID#1のノードの機器識別用タグTAGの値が0で、ノードID#2のノードの機器識別用タグTAGと同じ値0なので（ステップS330のNO）、ステップS340の処理に進む。

【0067】ステップS340に続くステップS350の処理では、ノードID#2のEntryFlagの値が1ではないので（ステップS340のNO）、Printerの機器識別用タグTAGの記述されているタグマップの機器識別用タグTAG#0を登録及び設定する（ステップS350のEntryFlag:=1→ステップS360）。

【0068】ステップS360に続くステップS370の処理では、CheckFlagの値を1に設定してステップS310に戻る。

【0069】ステップS370に続くステップS310の処理では、ノードID#2のノードのEntryFlag、CheckFlagの値が0のままなので、処理がステップS330の処理に進む（ステップS310のYES→ステップS320→ステップS330）。

【0070】ここで、ノードID#2の機器識別用タグTAGと同じ機器識別用タグTAGの値を持つノードは、ノードID#1のノードである。

【0071】ステップS340の処理では、ノードID#1のノードのEntryFlagの値が1なので（ステップS340のYES）、ステップS370の処理に進む。

【0072】ステップS340のYESに続くステップS370の処理では、ノードID#2のCheckFlagの値を1に設定して、ステップS310に再び戻る。

【0073】ステップS370に続くステップS310の処理では、ノードID#2のノードはまだEntry

11

Flagの値が0なので(ステップS310のYES)、処理がステップS320に移行する。

【0074】ステップS310に続くステップS320の処理では、CheckFlagの値が1(ステップS320のYES)なので、ステップS380へジャンプしてノードID#2のノードをタグキューに入れる。

【0075】ステップS380に続くステップS381では、CheckFlagの値が0のノードはないので(ステップS381のNO)、ステップS390の処理に進む。

【0076】図11に示すステップS391では、タグキューからノードID#2のノードを得る。ここでPrinterの機器識別用タグTAGの記述されているマップを検索して未使用の機器識別用タグTAG#1を小さい順から得る(ステップS391→ステップS392→ステップS393のNO)。

【0077】ステップS393のNOに続くステップS394の処理では、未使用の機器識別用タグTAG#1をノードID#2の機器識別用タグTAGに設定して、EntryFlagとCheckFlagの値を1に設定(EntryFlag:=1, CheckFlag:=1)してステップS390に戻る。

【0078】ステップS394に続くステップS390ではタグキューが空(ステップS390のYES)なので、処理は終了される。

【0079】一方、ステップS393において未使用タグがないと判定された場合(ステップS393のYES)、“未使用タグ無し”のエラー表示を行う処理が実行された後に終了する。

【0080】(第2実施形態)次に、図13を用いて第2実施形態について説明する。IEEE1394ネットワーク識別フローは、前述の図10のものを使用する。なお、第1実施形態において既に記述したものと同一の部分については、同一符号を付し、重複した説明は省略する。

【0081】図13は、前述の図9のIEEE1394ネットワークにさらにPrinterを2台追加接続したバスリセットが終了した状態を示している。

【0082】同図において、ステップS310の処理では、ノードID#0のノードでEntryFlagの値が0なので、EntryFlagの値が0のノードがあると判断され(ステップS310のYES)、ノードID#0のノードが検索される。

【0083】ステップS310のYESに続くステップS320の処理では、ノードID#0のノードのCheckFlagの値が0なので、ノードのCheckFlagの値が1でないと判断(ステップS320のNO)してステップS330の処理に進む。

【0084】ステップS320のNOに続くステップS330の処理では、ノードID#0と同じ機器識別用タ

12

グTAG#0を持つノードID#3のノードがあるので、ノードID#0と同じ機器識別用タグTAG#0を持つノードID#3のノードがあると判断(ステップS330のNO)してステップS340の処理に進む。

【0085】ステップS330のNOに続くステップS340の処理では、ノードID#3のノードのEntryFlagの値が1なので、ノードID#3のノードのEntryFlagの値が1と判断し(ステップS340のYES)、ステップS370の処理に進む。

10 【0086】ステップS370では、ノードID#0のノードのCheckFlagの値を1に設定して(CheckFlag:=1)、ステップS310に戻る。

【0087】ステップS370に続くステップS310の処理では、ノードID#0のノードはまだ、EntryFlagの値が0なので、ステップS320の処理に進む。

【0088】ステップS320の処理では、ノードID#0のノードのCheckFlagの値が1なので、ノードID#0のノードのCheckFlagの値が1であると判断し(ステップS320のYES)、ステップS380の処理に進み、ノードID#0をキューに入れる。

【0089】ステップS380に続くステップS381では、CheckFlagの値が0であるノード(ノードID#1のノード)があるので(ステップS381のYES)、再びステップS310に戻る(ステップS381→ステップS310)。

【0090】ステップS310の処理において、ノードID#1のノードに関してEntryFlag, CheckFlag共に0なので(ステップS310のYES→ステップS320のNO)、ステップS330の処理に進む。

【0091】ステップS320のNOに続くステップS330の処理では、ノードID#1と同じ機器識別用タグTAGを持つノードが存在しないので(ステップS330のNO)、ステップS350の処理に進む。

【0092】ステップS330のNOに続くステップS350の処理では、ノードID#1のノードのEntryFlagの値を1に設定する。

40 【0093】さらに、ステップS350に続くステップS360の処理では、ノードID#1の機器識別用タグTAG(このときの値=2)をPrinterの機器識別用タグTAGの記述されているマップに登録する。

【0094】ステップS360に続くステップS370では、CheckFlagの値を1に設定して、ステップS310に戻る。

【0095】ステップS370に続くステップS310の処理では、EntryFlagの値が0のノードはないので(ステップS310のNO)、ステップS390の処理に進む。

【0096】ステップS310のNOに続くステップS390の処理では、タグキューは空ではないので（ステップS390のNO）、ステップS391の処理が実行されてノードID#0の機器識別用タグTAGをタグキューから得る。

【0097】ステップS391に続くステップS392では、Printerの機器識別用タグTAGの記述されているマップを検索（ステップS392）して未使用の機器識別用タグTAG（値=4）を得る。

【0098】ステップS392に続くステップS393の処理では、未使用の機器識別用タグTAGがあるので（ステップS393のNO）、ステップS394の処理に進む。

【0099】ステップS393のNOに続くステップS394の処理では、ノードID#0のノードのEntryFlag、CheckFlagの値を1に設定してステップS390に戻る。

【0100】ステップS394に続くステップS390では、タグキューが空になったので（ステップS390のYES）、処理を終了する。

【0101】以上説明したように、第1実施形態または第2実施形態によれば、所定のカテゴリ内での同種類のネットワーク機器を識別するための機器識別用タグTAGに基づいて同種類のネットワーク機器の個々を識別するので、IEEE1394ネットワーク上に同種類のネットワーク機器を複数台接続する際、ベンダーが異なるIEEE1394ネットワーク間を相互接続するような場合、一方のIEEE1394ネットワークに接続されていたネットワーク機器を外して他方のIEEE1394ネットワークに接続するような場合、あるいはバスリセットが随時発生するような場合であっても、ノードID再設定前後でノードIDが変更されるようなケースを回避した一元的なノードID管理を実現できるようになり、その結果、ノードID再設定以前に使用してきたネットワーク機器が使用できなくなるケース、別の同種類のIEEE1394ネットワークに接続されるケースを回避できるようになる。

【0102】

【発明の効果】ネットワーク上に同種類のネットワーク機器を複数台接続する際、ベンダーが異なるネットワーク間を相互接続するような場合、一方のネットワークに接続されていたネットワーク機器を外して他方のネットワークに接続するような場合、あるいはバスリセットが随時発生するような場合であっても、ID再設定前後でIDが変更されるようなケースを回避した一元的なID管理を実現し、ID再設定以前に使用してきたネットワー

ク機器が使用できなくなるケース、別の同種類のネットワークに接続されるケースを回避できるようになる。

【図面の簡単な説明】

【図1】本発明のネットワーク機器識別方法の第1実施形態を説明するためのIEEE1394ネットワーク構成図である。

【図2】図1のIEEE1394ネットワークにスキャナとプリンタが追加接続された状態を示している。

【図3】第1実施形態のネットワーク機器識別方法の動作を説明するためのフローチャートである。

【図4】CSR中のコンフィグレーションROM内のデータフィールドであるNode_Unique_IDリーフのデータ構造を示す図である。

【図5】図4のNode_Unique_IDリーフ内のchip_id_hiに機器識別用タグを実装したデータ構造を説明するための図である。

【図6】図4のNode_Unique_IDリーフ内のchip_id_loに機器識別用タグを実装したデータ構造を説明するための図である。

【図7】図4のNode_Unique_IDリーフ内のchip_id_hiに機器識別用タグを実装した他のデータ構造を説明するための図である。

【図8】同じ種類のネットワーク機器の機器識別用タグを自動的に設定する処理を説明するためのフローチャートである。

【図9】同じ種類のネットワーク機器の機器識別用タグを自動的に設定する処理の他の実施形態のネットワーク構成図である。

【図10】バスリセットによる機器識別用タグの設定の動作を説明するためのフローチャートである。

【図11】バスリセットによる機器識別用タグの設定の動作を説明するためのフローチャートである。

【図12】64ビット長のタグマップレジスタのデータ構造を説明するための図である。

【図13】図9のIEEE1394ネットワークにさらにPrinterを2台追加接続しかつバスリセットが終了した状態を示している。

【符号の説明】

TAG…機器識別用タグ

40 TAG#…タグ番号

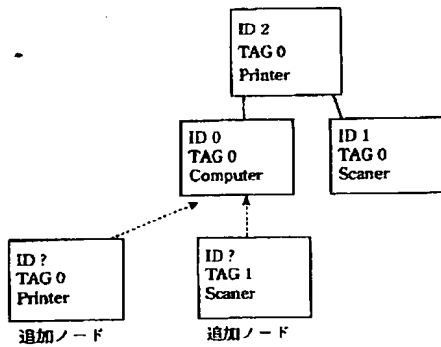
Printer…プリンター

Scanner…スキャナー

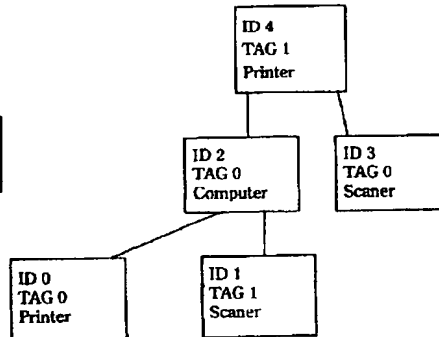
Computer…コンピュータ

CSR…Control Status Register

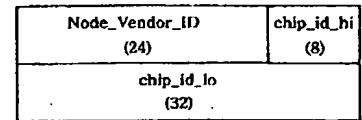
【図1】



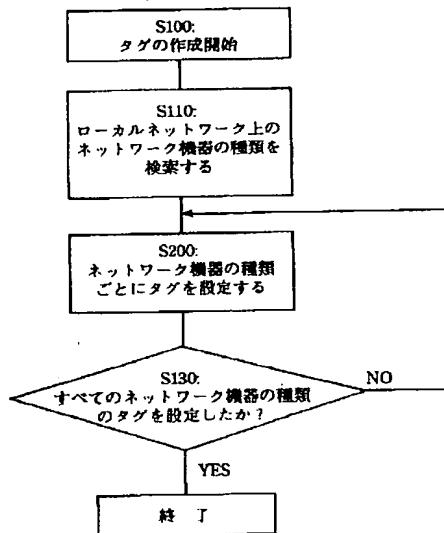
【図2】



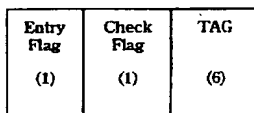
【図4】



【図3】



【図7】

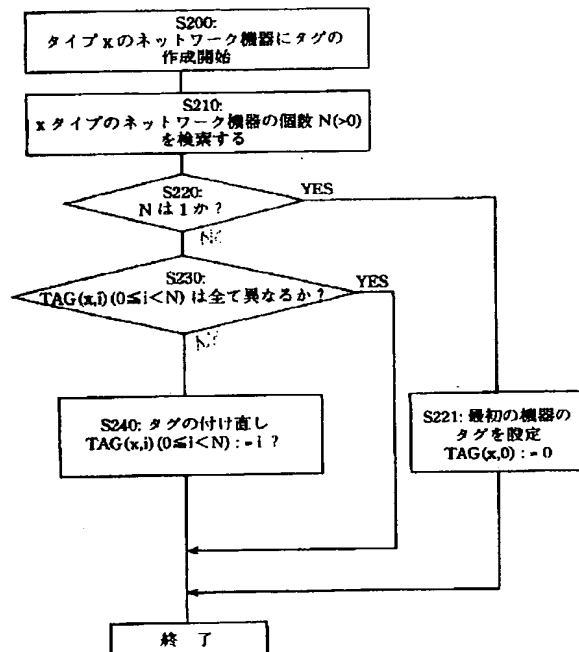


【図5】

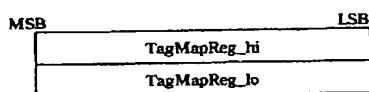


【図6】

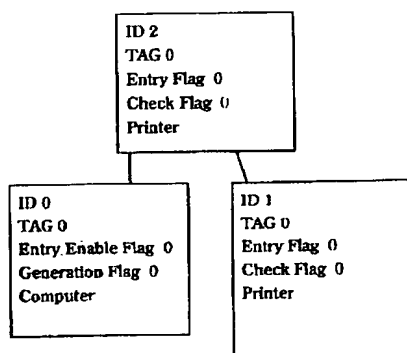
【図8】



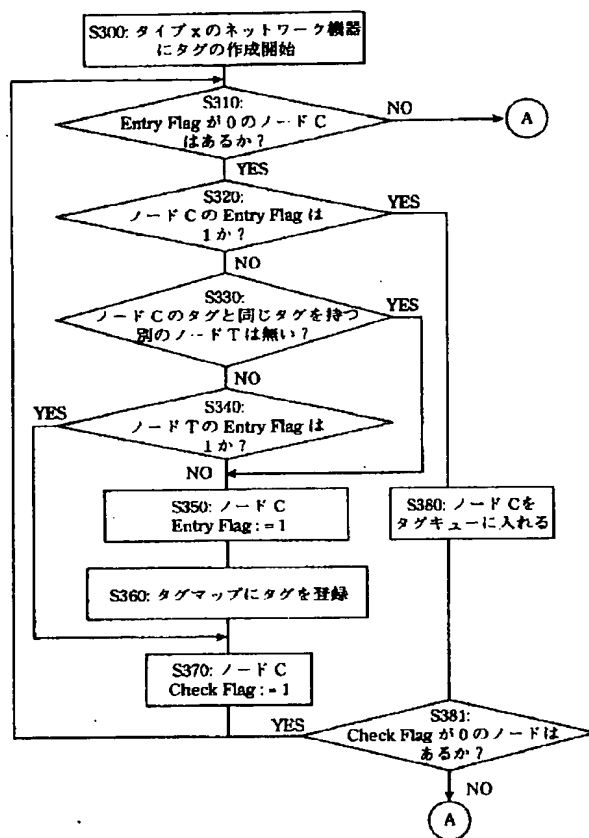
【図12】



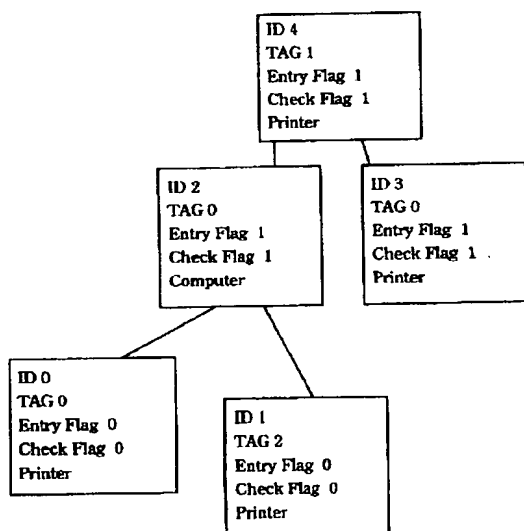
【図9】



【図10】



【図13】



【図11】

